

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Group Art Unit: TBA  
Examiner: Unassigned

JC986 U.S. PRO  
09/987039  
11/13/01

In re PATENT APPLICATION of

Commissioner of Patents  
Washington, D.C. 20231

Sir:

Submitted herewith is a certified copy of Applicant's first-filed Japanese Application No. 2000-354112 filed November 21, 2000, the rights of priority of which have been and are claimed pursuant to the provisions of 35 U.S.C. §119.

It is respectfully requested that receipt of this priority document be acknowledged.

Respectfully submitted,

November 13, 2001

Date

Steven M. Rabin - Reg. No. 29,102  
**RABIN & BERDO, P.C.**  
Telephone: 202-659-1915  
Telefax: 202-659-1898  
CUSTOMER NO. 23995

SMR:tlc

日本国特許庁  
JAPAN PATENT OFFICE

1099/98/039  
U.S. PRO  
11/13/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日  
Date of Application:

2000年11月21日

出願番号  
Application Number:

特願2000-354112

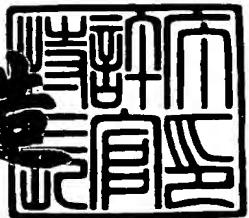
出願人  
Applicant(s):

沖電気工業株式会社

2001年 7月 9日

特許庁長官  
Commissioner,  
Japan Patent Office

及川耕造



出証番号 出証特2001-3063815

【書類名】 特許願  
 【整理番号】 KA003713  
 【提出日】 平成12年11月21日  
 【あて先】 特許庁長官 及川 耕造 殿  
 【国際特許分類】 G06F 15/78  
 G11C 16/02

## 【発明者】

【住所又は居所】 東京都港区虎ノ門1丁目7番12号 沖電気工業株式会社内

【氏名】 ▲高▼田 淑朗

## 【特許出願人】

【識別番号】 000000295

【氏名又は名称】 沖電気工業株式会社

## 【代理人】

【識別番号】 100086807

## 【弁理士】

【氏名又は名称】 柿本 恒成

## 【手数料の表示】

【予納台帳番号】 007412

【納付金額】 21,000円

## 【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9001054

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 マイクロコントローラ

【特許請求の範囲】

【請求項1】 実行命令コードで構成された実行プログラムが格納されたプログラム領域、前記プログラム領域中に修正すべき部分があるときにはこの部分の先頭の一部又は全部にプログラム抹消用の指定命令コードが上書きされたプログラム消滅領域、及び前記修正すべき部分の修正プログラムが格納された修正プログラム領域を有すると共に、割り込みプログラムが格納された再書き込み可能な不揮発性メモリと、

前記実行プログラムの実行時に、前記プログラム領域中の前記プログラム消滅領域を検出して割り込み信号を出力する検出手段と、

前記実行プログラムの前記実行命令コードを読み出し、この内容を解読して実行すると共に、前記実行プログラムの実行中に前記検出手段から前記割り込み信号が与えられると、前記割り込みプログラムに従い、前記実行プログラムの実行を中断して前記修正プログラムを実行し、前記修正プログラムの終了後に前記実行プログラムへ復帰する中央処理装置と、

を備えたことを特徴とするマイクロコントローラ。

【請求項2】 前記検出手段は、

前記実行プログラムの実行時に前記実行命令コードと予め設定された前記指定命令コードとを比較して両者が1回一致したとき又は所定回連続して一致したときには前記割り込み信号を出力する比較手段と、

前記命令実行コードのアドレスが格納され、前記割り込み信号が与えられると、前記中央処理装置の制御を前記プログラム領域から前記修正プログラム領域へジャンプさせるためのアドレスを出力するアドレスレジスタと、

を有することを特徴とする請求項1記載のマイクロコントローラ。

【請求項3】 前記不揮発性メモリ内には、前記プログラム消滅領域のアドレスを格納した固定アドレス格納領域が設けられ、

前記検出手段は、前記実行プログラムの初期化時に前記固定アドレス格納領域内の前記プログラム消滅領域のアドレスが格納されるアドレスレジスタと、前記

実行プログラムの実行時に前記実行命令コードのアドレスと前記アドレスレジスタに格納された前記プログラム消滅領域のアドレスとを比較して両者が一致したときには前記割り込み信号を出力する比較手段とを有することを特徴とする請求項1記載のマイクロコントローラ。

【請求項4】 実行命令コードで構成された実行プログラムが格納されたプログラム領域、前記プログラム領域中に修正すべき部分があるときにはこの部分の先頭の一部又は全部にプログラム抹消用の指定命令コードが上書きされたプログラム消滅領域、前記修正すべき部分の修正プログラムが格納された修正プログラム領域、前記修正プログラム領域のアドレスが格納された固定アドレス格納領域、及び飛び先判定プログラムを有すると共に、割り込みプログラムが格納された再書き込み可能な不揮発性メモリと、

前記実行プログラムの実行時に前記実行命令コードと予め設定された前記指定命令コードとを比較して両者が一致したときには割り込み信号を出力する比較手段と、

前記実行プログラムの前記実行命令コードを読み出し、この内容を解読して実行すると共に、前記実行プログラムの実行中に前記比較手段から前記割り込み信号が与えられると、前記割り込みプログラムに従い、前記実行プログラムの実行を中断して前記飛び先判定プログラムへ移り、前記飛び先判定プログラムにおいて前記固定アドレス格納領域を参照して前記修正プログラムの位置を判定し、前記修正プログラムを実行して該修正プログラムの終了後に前記実行プログラムへ復帰する中央処理装置と、

を備えたことを特徴とするマイクロコントローラ。

【請求項5】 個別の機能を実現する複数の個別プログラム、前記個別プログラムを統括するメインプログラム、前記メインプログラムの指示に従い前記複数の個別プログラム中のどの個別プログラムを実行するかを判断する飛び先判定プログラム、前記個別プログラムに修正の必要が生じた場合にその修正後の修正プログラムが格納された修正プログラム領域、及び飛び先のアドレスが格納された複数の飛び先リストを有する再書き込み可能な不揮発性メモリと、

前記メインプログラムを実行するときには、前記飛び先判定プログラムにおい

て前記飛び先リストを参照し、前記修正プログラムがない場合には該当する前記個別プログラムを実行し、前記修正プログラムがある場合には該当する前記修正プログラムを実行する中央処理装置と、

を備えたことを特徴とするマイクロコントローラ。

【請求項6】 前記不揮発性メモリは、E P R O Mで構成したことを特徴とする請求項1～5のいずれか1項に記載のマイクロコントローラ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、再書き込み可能な不揮発性メモリ（例えば、Erasable Programmable Read Only Memory；E P R O M）が搭載され、本来1回に限りプログラム変更可能なOTPマイクロコントローラ（One Time Programmable MicroController、以下「OTPマイコン」という。）、特にこのOTPマイコンに複数回プログラム変更できる機能を付加したマイクロコントローラ（以下「マイコン」という。）に関するものである。

【0002】

【従来の技術】

従来、中央処理装置（以下「C P U」という。）及び読み出し専用メモリであるR O M（Read Only Memory）等を内蔵したワンチップマイコンは、そのR O Mにプログラムを書き込み、その後はそのプログラムを変更することができないが、半導体の製造段階から例えばガラスマスクを用いて製造できるので、安価に大量生産できる特徴を有していた。これに対し、例えば、アプリケーションプログラム（応用プログラム）の開発者からは、開発終了後のプログラムのバグ（bug；プログラム中の誤り部分）に対処する機能を要求する声があった。そのバグに対応するためには、ガラスマスクの製造からやり直す必要があり、長期間待たされるばかりでなく、特に少量多品種の製品に対してはコスト的に合わなかつたからである。

【0003】

そこで、OTPマイコンといって、半導体チップの内部にE P R O Mを混載す

るマイコンが1970年代に開発された。E PROMは、紫外線等によって電荷の消去が可能であり、ソース・ドレイン間に負の高電圧を印加し、浮遊ゲート（floating gate）に電子を蓄積させることにより書き込みが可能な不揮発性メモリである。このE PROMでは、浮遊ゲートに紫外線等による光電効果によって電子を帯電させるため、例えば、紫外線等を透過する石英の窓を有し、これを支持するためにセラミック製のパッケージが必要であり、E PROM及びこれを搭載するマイコンは高価にならざるを得なかった。

#### 【0004】

しかし、OTPマイコンは、紫外線等による消去はチップの製造段階にとどめ、その後はE PROMへの書き込みだけができるようにしたマイコンであり、原則的に同一領域に対する再書き込みができない構造になっている。このような制限はあるものの、例えば、プラスチックでパッケージでき安価に製造できること、原則1回に限りの書き込みであるがアプリケーションプログラムをマイコンの製造の後において書き込めるところから、アプリケーションプログラムの開発者の要望に応える製品であり、現実に自動車の室温コントロール、コンピュータのキー・ボード制御、モータ制御、モデムの制御等の分野にOTPマイコンは広く普及している。

#### 【0005】

図2は、従来のOTPマイコンを示す一般的な構成図である。

このOTPマイコンは、通常のマイコンと同様に、半導体チップ1の内部に、プログラム格納用のE PROM2、E PROM書き込み回路3、データ格納用の隨時読み書き可能なメモリであるRAM(Random Access Memory)5、内部バス6、及びマイコン全体をプログラム制御するCPU7等を有している。OTPマイコンでは、ROMではなくE PROM2にプログラムが格納されている点が通常のマイコンと大きく異なる。E PROM2の書き込みには、書き込み回路3の他に、特別の高電圧端子4が設けられている。この高電圧端子4は、通常、電源電圧VDD(=5V)を印加するが、書き込み時には高電圧VPP(=12V程度)を印加してトンネル効果を起こすのに十分な電界を供給する。リセット時に、高電圧端子4を接地電圧VSS(=0V)にしておくと、E PROM2のテス

トモードに移行するといったOTPマイコンもある。

## 【0006】

## 【発明が解決しようとする課題】

図2のようなOTPマイコンの書き込み制限については、E PROM2の同一ゲートへの書き込みが物理的に制限されているということであって、未だ書き込みがされていないゲートへの書き込みができないということではない。そもそも物理的に見てE PROM2への書き込みとは、制御ゲートに高電圧VPPを印加し、浮遊ゲートの電荷をトンネル効果を用いて放電させることである。E PROM2のゲートは、一旦書き込みによって放電した後は紫外線等による帯電をしない限り再び電荷を蓄積することなく、逆に一旦紫外線照射等によって帯電すれば、その後書き込みがない限り半永久的に電荷を帯電し続ける。ただ、プログラムのコードはそのアドレスに従って連続的に並び、それに対応するゲート群も一定の領域を占めているから、その領域に新たな書き込みをすれば、一旦書き込まれたゲートは消えずに残り、新たなプログラムのコードに対応しなくなる。このため、全く書き込みのされていない領域への新たな書き込みは問題はないが、重複する書き込みはプログラムを破壊するだけである。さらに、新たな領域にプログラムを追加したとしても、どのように元のプログラムとリンクさせるのか定かではない。このような状況から、E PROM2は原則的に再書き込みができないものとされてきた。

## 【0007】

しかし、現実のアプリケーションプログラムの開発段階においては、E PROM2への再書き込みの要求が少なくない。例えば、アプリケーションプログラムの書き込みを行い、大量に製品を製造し、又は出荷したが、その後テスト又は客先でプログラムのバグが発見されたり、客先の要求に従ってプログラムの一部を改変したい場合である。かかる場合に、新たなプログラムを書き込んだOTPマイコンを製造することはもちろん可能であるが、既に前のプログラムを書き込んでしまった製品は全て無駄になってしまう。この損失は、プログラムの小さなバグが多発した場合には耐え難いものになるだろう。

## 【0008】

かかる場合に、マイコンの内部にアプリケーションプログラムを記憶したROMを有し、かつ、外部に修正プログラムを記憶したEPROMを有するコンピュータシステムで、修正プログラムをRAMに上書きする方法が知られている。例えば、特許第2995030号公報では、CPUがCPU割り込み回路において特定の割り込みアドレスを指定し、プログラムが走行中当該アドレスと一致した場合にはCPU割り込みを起こし、その割り込みプログラムにおいてRAMに対し修正プログラムの上書きを行う。しかし、この方法では、外部に修正プログラムを置くEPROMを必要とし、何らOTPマイコン固有の特性を利用していないばかりか、OTPマイコンの利便性や経済性を損なう構成である。

#### 【0009】

本発明は、前記従来技術が持っていた課題を解決するために、EPROMではゲートレベルにおいて未だ書き込まれていないゲートに対し書き込みが可能であるという点に着目し、若干の電子回路の追加と新たなプログラムの追加によって、OTPマイコンに複数回プログラム変更できる機能を付加したマイコンを提供することを目的とする。

#### 【0010】

##### 【課題を解決するための手段】

前記課題を解決するために、本発明のうちの第1の発明は、マイコンにおいて、実行命令コードで構成された実行プログラムが格納されたプログラム領域、前記プログラム領域中に修正すべき部分があるときにはこの部分の先頭の一部又は全部にプログラム抹消用の指定命令コードが上書きされたプログラム消滅領域、及び前記修正すべき部分の修正プログラムが格納された修正プログラム領域を有すると共に、割り込みプログラムが格納された再書き込み可能な不揮発性メモリと、前記実行プログラムの実行時に、前記プログラム領域中の前記プログラム消滅領域を検出して割り込み信号を出力する検出手段と、前記実行プログラムの前記実行命令コードを読み出し、この内容を解読して実行すると共に、前記実行プログラムの実行中に前記検出手段から前記割り込み信号が与えられると、前記割り込みプログラムに従い、前記実行プログラムの実行を中断して前記修正プログラムを実行し、前記修正プログラムの終了後に前記実行プログラムへ復帰するC

PUとを備えている。

【0011】

第2の発明は、第1の発明のマイコンにおいて、前記検出手段は、前記実行プログラムの実行時に前記実行命令コードと予め設定された前記指定命令コードとを比較して両者が1回一致したとき又は所定回連続して一致したときには前記割り込み信号を出力する比較手段と、前記命令実行コードのアドレスが格納され、前記割り込み信号が与えられると、前記CPUの制御を前記プログラム領域から前記修正プログラム領域へジャンプさせるためのアドレスを出力するアドレスレジスタとを有している。

【0012】

第3の発明は、第1の発明のマイコンにおいて、前記不揮発性メモリ内には、前記プログラム消滅領域のアドレスを格納した固定アドレス格納領域が設けられ、前記検出手段は、前記実行プログラムの初期化時に前記固定アドレス格納領域内の前記プログラム消滅領域のアドレスが格納されるアドレスレジスタと、前記実行プログラムの実行時に前記実行命令コードのアドレスと前記アドレスレジスタに格納された前記プログラム消滅領域のアドレスとを比較して両者が一致したときには前記割り込み信号を出力する比較手段とを有している。

【0013】

第4の発明は、マイコンにおいて、実行命令コードで構成された実行プログラムが格納されたプログラム領域、前記プログラム領域中に修正すべき部分があるときにはこの部分の先頭の一部又は全部にプログラム抹消用の指定命令コードが上書きされたプログラム消滅領域、前記修正すべき部分の修正プログラムが格納された修正プログラム領域、前記修正プログラム領域のアドレスが格納された固定アドレス格納領域、及び飛び先判定プログラムを有すると共に、割り込みプログラムが格納された再書き込み可能な不揮発性メモリと、前記実行プログラムの実行時に前記実行命令コードと予め設定された前記指定命令コードとを比較して両者が一致したときには割り込み信号を出力する比較手段と、前記実行プログラムの前記実行命令コードを読み出し、この内容を解読して実行すると共に、前記実行プログラムの実行中に前記比較手段から前記割り込み信号が与えられると、

前記割り込みプログラムに従い、前記実行プログラムの実行を中断して前記飛び先判定プログラムへ移り、前記飛び先判定プログラムにおいて前記固定アドレス格納領域を参照して前記修正プログラムの位置を判定し、前記修正プログラムを実行して該修正プログラムの終了後に前記実行プログラムへ復帰するCPUとを備えている。

#### 【0014】

このような構成を採用したことにより、実行プログラム（例えば、アプリケーションプログラム）の修正すべき部分の先頭の一部又は全部をプログラム消滅領域に変え（例えば、全て書き込み状態のゲートに変え）、これを検出手段で検出するようにし、さらに、CPUが修正後のプログラムを実行するようにこの修正後のプログラムの追加書き込みを行うようにしている。そして、実行プログラムの実行時には、検出手段によってプログラム消滅領域が検出され、プログラム制御が追加の修正プログラムに移る。追加の修正プログラムに移ると、このプログラムが実行され、この実行後に元の実行プログラムへ復帰する。

#### 【0015】

第5の発明は、マイコンにおいて、個別の機能を実現する複数の個別プログラム、前記個別プログラムを統括するメインプログラム、前記メインプログラムの指示に従い前記複数の個別プログラム中のどの個別プログラムを実行するかを判断する飛び先判定プログラム、前記個別プログラムに修正の必要が生じた場合にその修正後の修正プログラムが格納された修正プログラム領域、及び飛び先のアドレスが格納された複数の飛び先リストを有する再書き込み可能な不揮発性メモリと、前記メインプログラムを実行するときには、前記飛び先判定プログラムにおいて前記飛び先リストを参照し、前記修正プログラムがない場合には該当する前記個別プログラムを実行し、前記修正プログラムがある場合には該当する前記修正プログラムを実行するCPUとを備えている。

#### 【0016】

このような構成を採用したことにより、CPUによるメインプログラムの実行時には、飛び先判定プログラムにおいて飛び先リストが参照され、この飛び先リスト内に修正プログラムがない場合には該当する個別プログラムが実行され、修

正プログラムがある場合には該当する修正プログラムが実行される。

第6の発明は、第1～第5の発明のいずれかのマイコンにおいて、不揮発性メモリをE PROMで構成している。

### 【0017】

#### 【発明の実施の形態】

##### （実施形態の原理）

図3は、本発明の実施形態を示すOTPマイコンの概略の構成図である。

このOTPマイコンでは、半導体チップ11内に、再書き込み可能な不揮発性メモリ（例えば、E PROM）12、高電圧端子14に接続されたE PROM書き込み回路13、RAM15、内部バス16、CPU17、及び検出手段18等を有している。E PROM12、RAM15、CPU17、及び検出手段18は、内部バス16を介して相互に接続されている。

### 【0018】

E PROM12は、実行命令コードで構成された実行プログラム（例えば、アプリケーションプログラム）が格納されたプログラム領域と、該プログラム領域中に修正すべき部分があるときにはこの部分の先頭の一部又は全部にプログラム抹消用の指定命令コードが上書きされたプログラム消滅領域と、その修正すべき部分の修正プログラムが格納された修正プログラム領域と、割り込みプログラム等が格納された領域とを有している。高電圧端子14には、通常、電源電圧VDDが印加され、書き込み時には高電圧VPPが印加され、E PROM書き込み回路13に与えられる。E PROM書き込み回路13は、高電圧VPPを用いてE PROM12に対して命令コード等を書き込む回路である。CPU17は、アドレスを選択してE PROM12内のプログラムを読み出し、この内容を解読して実行する装置である。検出手段18は、CPU17によるE PROM12内のアプリケーションプログラムの実行時に、該E PROM12内のプログラム消滅領域を検出して割り込み信号S18を出力し、CPU17に与えるものである。

### 【0019】

図4は、図3中のCPU17の概略を示す構成図である。

このCPU17は、演算部21、制御部22、及びレジスタ部23等を有し、

これらが内部バス16に接続されている。演算部21は、算術論理演算ユニット(AlU)等を有し、データの演算を行う機能を有している。制御部22は、マイコン全体を制御するための各種の制御信号を出力するものであり、EPROM12内の命令コードを保持する命令レジスタ22a、該命令レジスタ22aから出力された命令コード等を解読する命令デコーダ22b、及びタイミング/制御信号発生回路22c等を有している。タイミング/制御信号発生回路22cは、命令デコーダ22bの出力信号、割り込み信号S18、及びクロック信号等を入力し、割り込み制御信号等の各種の制御信号を出力する回路である。レジスタ部23は、EPROM12のメモリアドレスを出力するものであり、データを保持する複数のレジスタからなるレジスタ群23a、スタックポインタ23b、及びメモリアドレスを出力するプログラムカウンタ23c等を有している。

#### 【0020】

図1は、本発明の実施形態の原理を示す概念図である。

EPROM12のアドレス空間31は、書き込みのされたアプリケーションプログラム領域32、34と、書き込みのされていない空白領域37とに分けることができる。空白領域37は、修正プログラムを追加する都合上必要な領域である。

#### 【0021】

例えば、アプリケーションプログラムの開発終了後にプログラムの一部にバグが発見され、これを修正したいと考えたとする。しかし、このままでは空白領域37に追加のプログラムを書き込んだとしても、元のプログラムとのリンクを取ることはできない。なぜなら、アプリケーションプログラム領域32、34は、自ら閉じていて追加の修正プログラム領域35に制御を渡す手段を有していないからである。OTPマイコンが原則的に再書き込みができないとされる理由もある。

#### 【0022】

そこで、本発明の実施形態では、アプリケーションプログラム領域32、34内のバグを解消するために、プログラムの修正すべき部分の先頭の一部又は全部にプログラム抹消用の指定命令コードが上書きされた（例えば、全て書き込み状

態のゲートに変えた) プログラム消滅領域33を作り、このプログラム消滅領域33を検出するための検出手段18を追加し、さらに修正プログラム領域35を作ってこれを実行できるような構成にしている。

#### 【0023】

即ち、CPU17によるEPROM12内のアプリケーションプログラムの実行時には、検出手段18により、前段のアプリケーションプログラム領域32の終端でプログラム消滅領域33における全ビット書き込みの状態を検出し、割り込み信号S18を出力してCPU17に与える。CPU17は、EPROM12に格納されたCPU割り込みプログラムに従い、プログラム制御を追加の修正プログラム領域35へ移す(①)。CPU17は、CPU割り込みプログラムに従い、追加の修正プログラム領域35内の修正プログラムを実行し、これが終了すると、後段のアプリケーションプログラム領域34へ復帰し(②)、このアプリケーションプログラム領域34内のプログラムを実行する。

よって、本実施形態では、OTPマイコンという名称に拘わらず、複数回プログラム書き込みを行うことができる効果を有している。

#### 【0024】

##### (第1の実施形態)

図1の原理を実現するために、第1の実施形態では、図1及び図3に示す検出手段18を、例えば、ROM割り込み回路で構成している。

EPROM12においては、追加の書き込みにより全てのゲートを書き込み状態にすることができます。そのため、EPROM12に格納された命令コードをこの論理に従い、このバイト値を「00」又は「ff」にすることができる。通常、EPROM12においては、書き込み状態にしたゲートを論理値「0」に対応させるから、このコードは「00」になるだろう。ところで、多くのマイコンにおいては、コード「00」はNOP(No Operation)命令に割り当てている。これは、バグを修正するデバッグ(debug)の段階で誤った命令コードを後にNOP命令に変更するのに都合がよい。NOP命令は、何らCPU17の状態を変えないからである。しかし、このようなマイコンにおいて、NOPを検出する毎に割り込みを生じさせるのは現実的ではない。事実上NOP命令が使えなくなるか

、あるいはN O P命令のコードを「00」以外に変更せざるを得ないからである。そこで、本実施形態では、検出手段18によって命令コード「00」を連続して（又は1つ）検出したときに割り込み信号S18を発生させるような汎用性有る構成を採用している。

#### 【0025】

図5は、本発明の第1の実施形態を示すものであって図3中の検出手段18の構成図である。

この検出手段18は、命令レジスタ22aに置かれた、又はパイプラインで実行段階にある実行命令コード41と、予め設定された指定命令コード42（例えば「00」）とを比較する比較回路43を有し、この出力端子に信号反転用のインバータ44が接続されている。また、比較回路43の出力端子には、カウンタ45の入力端子INが接続されると共に、このカウンタ45のリセット端子Rがインバータ44の出力端子に接続されている。カウンタ45の出力端子からは割り込み信号S18が出力され、この割り込み信号S18がCPU17内のタイミング／制御信号発生回路22cに与えられると共に、1段又は複数段のアドレスレジスタ46に与えられるようになっている。アドレスレジスタ46は、シフトレジスタ等で構成され、プログラム消滅領域33の先頭アドレスを保持し、割り込み信号S18が与えられると、この保持したプログラム消滅領域33の先頭アドレスを出力して内部バス16へ送る機能を有している。ここで、比較回路43、インバータ44及びカウンタ45により、比較手段が構成されている。

#### 【0026】

以下、図5の動作を説明する。

CPU17によって図1のアプリケーションプログラム領域32のアプリケーションプログラムが実行されると、このアプリケーションプログラムの実行命令コード41が比較回路43に入力される。比較回路43では、入力された実行命令コード41と、予め設定された指定命令コード42（例えば「00」）とを比較し、両入力が一致する場合には、一致信号（キャリー）を出力し、カウンタ45のカウント値を増加させる。比較回路43の比較結果が不一致の場合には、この不一致の信号がインバータ44で反転されてカウンタ45のリセット端子Rに

入力され、該カウンタ45がリセットされる。

【0027】

カウンタ45が例えば3ビット構成の場合、指定命令コード42（例えば「00」）が8回連続したとき、このカウンタ45の出力端子OUTからキャリーが出力され、これが割り込み信号S18としてCPU17内のタイミング／制御信号発生回路22cに与えられると共に、アドレスレジスタ46に与えられる。割り込み信号S18がアドレスレジスタ46に与えられると、このアドレスレジスタ46からプログラム消滅領域33の先頭アドレスが出力され、内部バス16へ送られる。

【0028】

割り込み信号S18が与えられたCPU17は、EPROM12内のCPU割り込みプログラムに従い、アプリケーションプログラムの設計段階から予め定めていた修正プログラム領域35の先頭アドレスへジャンプし（図1の①）、この修正プログラムを実行する。プログラム消滅領域33の最終アドレス（即ち、後段のアプリケーションプログラム領域34の先頭アドレス）は、該プログラム消滅領域33を作った段階で予め分かっている。CPU17は、修正プログラム領域35の修正プログラムを実行した後、後段のアプリケーションプログラム領域34の先頭アドレスに復帰し（図1の②）、このアプリケーションプログラムを連続して実行する。

【0029】

次に、この第1の実施形態が、複数回の修正に耐え得る柔軟性を有することについて説明する。

図5の検出手段18では、CPU17による後段のアプリケーションプログラム領域34のアプリケーションプログラムを実行しているときに、次のプログラム消滅領域を検出した場合には、CPU割り込みプログラムにおいてその検出したアドレスを比較することにより、次の第2の修正プログラム領域にジャンプすることができる。アドレスレジスタ46は、複数のプログラム消滅領域のアドレスを保持する機能を果たし、カウンタ45から出力される割り込み信号S18によってそのときのアドレスを内部バス16からCPU17に取り込むことにより

、上記目的（複数回の修正）を達成できる。この目的に合うように、アドレスレジスタ46は、シフトレジスタ等の構成を持つことが望ましい。割り込みプログラムに従ってCPU17は、アドレスレジスタ46に置かれたアドレス情報をチェックしに行くことになる。この際、アドレスの番地の大きさの順番を指標にして何番目の修正プログラムを実行すべきか判断できる。

#### 【0030】

以上のように、この第1の実施形態では、次の（a）、（b）のような効果がある。

（a） 本実施形態では、EEPROM12に重ね書きができる点に着目し、指定命令コード42（例えば「00」）を作りだし、かつ、これを検出手段18で検出することでCPU17に対して割り込みを生じさせ、修正プログラムの実行を可能にしている。従来、外付けのEEPROMを用いてRAM15にプログラムを重ね書きする技術は存在したが、本実施形態のように予め修正を想定してその検出手段18を設け、何ら外付けの修正手段を使わずに、かつ、OTPマイコンの利便性を損なうことなくアプリケーションプログラムの修正を実現することができる。

#### 【0031】

（b） 命令コード「00」がNOP命令に対応することが多いことに鑑み、カウンタ45を用い、命令コード「00」を連続して検出した場合にのみCPU17に対して割り込みを生じさせる汎用性有る構成を採用している。この結果、OTPマイコンという名称に拘わらず、複数回のプログラム修正を可能にしたのであり、それに要するコストも若干の回路付加で済ませることができる。

#### 【0032】

この第1の実施形態では、次の（i）～（iii）のような変形例が考えられる。

（i） 命令コード「00」が特定の命令に割り当てられていないマイコンにおいては、図5のカウンタ45は不用となり、比較回路43の一致出力がそのまま割り込み信号S18として出力される構成を取ることもできる。

#### 【0033】

（ii） 指定命令コード42は理論上「00」に限られない。例えば、通常の命

令は最上位ビットを1にし、指定命令コード42は最上位ビットを0にすることによって、理論上区別することができ、かつ、EEPROM12の重ね書きも可能である。ただ、このような方式は命令コードの数を減少させるので、OTPマイコンの用途等に応じて適宜適用すべきである。

## 【0034】

(iii) 修正すべき回数が1回で十分な場合には、アドレスレジスタ46は必ずしも必要ではない。アドレスレジスタ46を設けない場合は、CPU割り込みプログラムは、予め固定された修正プログラム領域35に制御を移すことになる。

## 【0035】

## (第2の実施形態)

図6は、本発明の第2の実施形態を示すものであって図3中の検出手段18の他の構成図である。

この検出手段18では、図5の構成と異なり、CPU17の実行段階にある命令コードのEEPROM12上のアドレス47と、EEPROM12上の特定アドレス48とを、比較手段（例えば、比較回路）43で比較し、それらのアドレス47と48が一致する場合に該比較回路43から割り込み信号S18を出力し、CPU17に割り込みを生じさせるようになっている。特定アドレス48は、アドレスレジスタ49に保持されている。

## 【0036】

図7は、本発明の第2の実施形態を示すEEPROM12のアドレス空間の図である。

このEEPROM12のアドレス空間31は、図1と同様に、アプリケーションプログラム領域32、34、プログラム消滅領域33、及び追加の修正プログラム領域35の他に、固定アドレス格納領域51を有している。固定アドレス格納領域51は、プログラム消滅領域33のアドレスを格納する領域であり、CPU17によってデータ格納領域として参照される。CPU17が、アプリケーションプログラム領域32のアプリケーションプログラムを実行する前に、まず、アプリケーションプログラムの初期化を行い、この初期化の際に、固定アドレス格納領域51に格納されたプログラム消滅領域33のアドレスを、アドレスレジス

タ49に格納した後、アプリケーションプログラムを実行するようになっている。

【0037】

即ち、E PROM12の特定アドレス48は、プログラム消滅領域33を定めたものであり、この消滅領域33を定める際にそのアドレスが決定されるのであるから、予め定めることはできず、追加の修正プログラム領域35を作る際にE PROM12へ追加の書き込みを行うべきものである。本実施形態では、E PROM12のアドレス空間31において、新たに固定アドレス格納領域51を設け、特定アドレス48を格納する領域として使用する。なお、この固定アドレス格納領域51は、CPU17によってあくまでデータ格納領域として参照されるだけであるから、そのデータの参照及び格納において、比較回路43から出力される割り込み信号S18によってCPU割り込みが生じることはない。

【0038】

次に、図6及び図7の動作を説明する。

CPU17がE PROM12内のアプリケーションプログラムを実行する前に、このアプリケーションプログラムの初期化を行い、固定アドレス格納領域51に格納されたプログラム消滅領域33のアドレスがアドレスレジスタ49に格納される。CPU17がアプリケーションプログラム領域32のアプリケーションプログラムを実行すると、この実行命令コードのアドレス47と、アドレスレジスタ49に格納されたプログラム消滅領域33のアドレス（即ち、E PROM12の特定アドレス48）とが、比較回路43で比較され、それらのアドレス47と48が一致するときには該比較回路43から割り込み信号S18が出力され、CPU17へ与えられる。CPU17は、割り込み信号S18が与えられると、E PROM12内の割り込みプログラムに従い、前段のアプリケーションプログラム領域32から追加の修正プログラム領域35へ移り、この修正プログラム領域35の修正プログラムを実行する。修正プログラムの実行後、CPU17は割り込みプログラムに従い、後段のアプリケーションプログラム領域34へ復帰し、このアプリケーションプログラムを実行する。

【0039】

ここで、本実施形態のように固定アドレス格納領域51を設けるのではなく、レイアウト上直接、特定アドレス値の出力をE PROM12から引き出し、かつ比較回路43への入力にできないか考えてみると、これは現在のチップ上のE PROM12のレイアウト構成からして難しいと言わざるを得ない。チップ上に別個にE PROMを設ける手段も考えられるが、チップ面積の増加や端子数の増加等といったOTPマイコンの利点を失わせることになるだろう。

#### 【0040】

このような問題を解決するため、本実施形態では、アドレスレジスタ49を設け、アプリケーションプログラムの初期化の段階において、このアドレスレジスタ49に、固定アドレス格納領域51に格納したプログラム消滅領域33のアドレス（即ち、特定アドレス48）を移し、比較回路43への入力としている。特定アドレス48が何ら書き込みのない最終番地（例えば、2KバイトのE PROM12なら「0x7fff」）なら修正プログラム領域35はなかったことになるから、比較回路43を常に割り込み禁止となるように構成できるし、その特定アドレス48が最終番地以外なら追加の修正プログラム領域35があったと判断できるだろう。

#### 【0041】

本実施形態では、複数回のプログラム修正を以下のようにして達成することができる。

前回の追加の修正プログラム領域35のアドレスを修正する場合には、そのアドレスに重ね書きしてアドレスの値を0（例えば、2KバイトのE PROM12なら「0x000」）にしておき、新たに次の番地に追加の修正プログラム領域のアドレスを書き込んでおく。アプリケーションプログラムの初期化の段階において、その初期化プログラムは、固定アドレスが0の場合にはそのアドレスを無視し、次のアドレスを検査しに行き、固定アドレスが最終アドレス又は値が0でない場合にのみそれを修正プログラム領域のアドレスとして、アドレスレジスタ49に格納する。要はアプリケーションプログラムの初期化の段階において、予めプログラム修正を前提とした初期化プログラムを作つておけば、複数回のプログラム修正を容易に実現できる。

## 【0042】

アプリケーションプログラムの修正箇所が複数になった場合には、この数に応じてアドレスレジスタ49を複数段にすると共に、その数に応じて比較回路43の回路数を設ければよい。このように回路量が増えるが、修正箇所が複数でも対応可能である。

このように、この第2の実施形態では、次の(a)、(b)のような効果がある。

## 【0043】

(a) 本実施形態では、EPROM12の特定アドレス48を検出する手段を付加している。即ち、予め定められた固定アドレス格納領域51にその検出すべき特定アドレス48を書き込むことで、修正プログラムの存在を検知し、かつ、CPU17に対し割り込みを生じさせる比較回路43を設けている。この結果、アプリケーションプログラムの初期化段階で、プログラムの修正を前提とした初期化を行っておけば、その後は複数回の修正プログラムの追加に対応することができる。この場合には、NOP命令との衝突も起こらないので、第1の実施形態よりもプログラム作成段階での自由度は高い。ただ、本実施形態では、複数箇所のプログラム修正の場合に、修正に要する回路量が比例して増大する欠点がある。この点についてだけ着目すれば、第1の実施形態のアドレスレジスタ46を使う方が経済的である。この場合は、単にアドレスレジスタ46の段数を増やすだけで済むからである。

## 【0044】

(b) 従来、CPUが直接割り込みアドレスを指定し、このアドレスに至ると割り込みが生じるという装置があった。しかし、本実施形態では、固定アドレス格納領域51を持ち、これにプログラム消滅領域33のアドレスを重ね書きするのであるから、前記従来技術と異なる新しい技術であり、従来技術にはない上述したような効果がある。

## 【0045】

(第3の実施形態)

図8は、本発明の第3の実施形態を示すものであって図3中の検出手段18の

他の構成図である。

この検出手段18では、図5中のインバータ44、カウンタ45及びアドレスレジスタ46が省略され、実行命令コード41と指定命令コード42（例えば「00」）とを比較してこれらが一致する場合に割り込み信号S18を出力する比較手段（例えば、比較回路）43のみを有している。

#### 【0046】

図9は、本発明の第3の実施形態を示すE PROM12のアドレス空間の図である。

このE PROM12のアドレス空間31には、アプリケーションプログラム領域32、34、プログラム消滅領域33、及び追加の修正プログラム領域35の他に、飛び先判定プログラム61と、追加の修正プログラム領域35のアドレスが格納された固定アドレス格納領域62とを有している。また、このE PROM12には、CPU割り込みプログラム（図中省略）も格納されている。

#### 【0047】

次に、図8及び図9の動作を説明する。

図3のCPU17がE PROM12内のアプリケーションプログラム領域32のアプリケーションプログラムを実行すると、このアプリケーションプログラムの実行命令コード41と、予め設定された指定命令コード42（例えば「00」）とが、比較回路43により比較され、両者が一致する場合には割り込み信号S18が出力され、CPU17に与えられる。これにより、CPU17により実行される割り込みプログラムは、プログラム消滅領域33の存在を検知できる。本実施形態では、プログラム消滅領域33及び追加の修正プログラム領域35の確定につき、割り込みプログラムではなく、専ら飛び先判定プログラム61に依存していることを特徴とする。

#### 【0048】

即ち、CPU17が、前段のアプリケーションプログラム領域32のアプリケーションプログラムを実行中に、比較回路43がプログラム消滅領域33を検出すると、割り込み信号S18によってCPU17が、割り込みプログラムに従い飛び先判定プログラム61に制御を移す（③）。飛び先判定プログラム61は、

固定アドレス格納領域62を参照して(④)、追加の修正プログラム領域35の位置を判定し、制御を該追加の修正プログラム領域35の修正プログラムに移す(⑤)。これにより、新たな修正プログラムの実行が確保されたことになる。さらに、修正プログラム領域35の修正プログラムは、この実行の最終段階で、元のアプリケーションプログラムに復帰すべく後段のアプリケーションプログラム領域34へ制御を移し(⑥)、アプリケーションプログラムを実行する。これにより、追加の修正プログラムの実行が確保されたことになる。

#### 【0049】

このような修正プログラムの実行確保の方法は、修正プログラムに制限を生ずることに注意する必要がある。つまり、修正プログラム領域35の修正プログラムは、メインプログラム（アプリケーションプログラム領域32、34のアプリケーションプログラム）とは独立したものでなければならず、例えば、メインプログラム上にあるJUMP命令等で当該修正領域にリンクする場合には特別の考慮が必要になる。これは追加の修正プログラムは、それがEPROM12に置かれるという制限下で、あくまでプログラム消滅領域33以外のメインプログラムを変更できないという前提に立っているからである。仮にメインプログラムから当該修正プログラム領域35にジャンプする命令が複数あったとするとどうなるかを考えてみれば明らかである。そこではジャンプ先のプログラムは消滅している。何れのJUMP命令の場合も修正領域が検出され、割り込みプログラムに制御が移って、その結果修正プログラムが実行されるが、そこでは何れのJUMP命令からリンクされたかの判断ができないだろう。この困難のために当該修正プログラム領域35が必要以上に大きくなる可能性がある。

#### 【0050】

この第3の実施形態では、次のような効果がある。

本実施形態では、EPROM12のアドレス空間31において、独立した飛び先判定プログラム61と固定アドレス格納領域62とを有し、アプリケーションプログラム作成上の複雑さは増加するが、図5のアドレスレジスタ46あるいは図6のアドレスレジスタ49を別個に設ける必要がないので、第1及び第2の実施形態に比べて構成を簡易にできる。

## 【0051】

## (第4の実施形態)

図10 (A)、(B) は本発明の第4の実施形態を示すアプリケーションプログラムのプログラム構造図であり、同図 (A) はアプリケーションプログラムのアドレス空間を示す図、同図 (B) は同図 (A) 中の飛び先リストの例を示す図である。

## 【0052】

本実施形態のOTPマイコンでは、図1及び図3の検出手段18を設けずに、EEPROM12に格納されるアプリケーションプログラムによって複数回のプログラム修正を可能にしている。

図10 (A) に示すように、EEPROM12に格納されるアプリケーションプログラムのアドレス空間には、メインプログラム71、飛び先判定プログラム72、飛び先リスト( $L_1, \dots, L_n$ )73、個別プログラム( $K_1, \dots, K_n$ )74、及び修正プログラム領域( $S_1, \dots, S_m$ )75が設けられている。メインプログラム71は、個別の機能を実現する個別プログラム( $K_1, \dots, K_n$ )74を統括する役割を持つ。個別プログラム( $K_1, \dots, K_n$ )74は、n個( $K_1, \dots, K_n$ )のプログラムの集まりであり、各プログラムは例えば通信プログラム、表示プログラム、関数計算プログラム等、個別の機能を実現している。

## 【0053】

メインプログラム71の内部又はこれとは別個に、飛び先判定プログラム72が設けられ、これはメインプログラム71の指示に従ってどの個別プログラム( $K_1, \dots, K_n$ )74を実行するかを判断するプログラムである。この判断には、飛び先リスト( $L_1, \dots, L_n$ )73が用いられる。飛び先リスト( $L_1, \dots, L_n$ )73は、各個別プログラム( $K_1, \dots, K_n$ )74に対応してn個( $L_1, \dots, L_n$ )ある。修正プログラム領域( $S_1, \dots, S_m$ )75は、個別プログラム( $K_1, \dots, K_n$ )74に修正の必要が生じた場合にこの修正後のプログラムを格納する領域であり、ここではm個( $S_1, \dots, S_m$ )の修正プログラム領域75が設けられている。

## 【0054】

飛び先リスト ( $L_1, \dots, L_n$ ) 73 は、図9の修正プログラム領域35のアドレスを格納する固定アドレス格納領域62に対応している。図7の固定アドレス格納領域51は、プログラム消滅領域33のアドレスを格納する領域であり、本実施形態の飛び先リスト ( $L_1, \dots, L_n$ ) 73 と使用目的が異なっている。

以下、図10 (B) の飛び先リストの例を参照しつつ、どのようにして複数回のプログラム修正を実現するかを説明する。

#### 【0055】

図3のCPU17が実行するメインプログラム71において、個別の機能（例えば、文字表示）を実現したいと考えたとき、メインプログラム71はその機能を実現する個別プログラム（例えば、 $K_i, 1 \leq i \leq n$ ）に制御を渡すべく、飛び先判定プログラム72を該パラメータ  $i$  ( $1 \leq i \leq n$ ) と共に呼び出す。すると、飛び先判定プログラム72は、そのパラメータ  $i$  に従い、飛び先リスト  $L_i$  を参照しに行き、この飛び先リスト  $L_i$  に従って飛び先を判定する。この飛び先是、何ら修正がない場合には個別プログラム  $K_i$  になるし、修正プログラムの場合には修正プログラム領域 ( $S_1, \dots, S_m$ ) 75の1つ（例えば、 $S_j, 1 \leq j \leq m$ ）になる。

#### 【0056】

図10 (B) の飛び先リスト73において、飛び先リスト  $L_1$  を参照した場合が何ら修正がない場合に当たる。ちなみに飛び先リスト  $L_1$  では、2KバイトのE PROM12の最終アドレス「0x7fff」を前提として、リストの最初に飛び先 (A1) があり、これは個別プログラム (K1) の先頭アドレスに一致することになる。最初に存在する飛び先 (A1) は、これがアドレス「0x000」でないことから、修正プログラムはないと判断される。修正プログラムがある場合には、E PROM12が重ね書きできるという本発明の原理に従い、この飛び先是「0x000」に書き換えられているはずである。ここでは、飛び先判定プログラム72は飛び先リスト73の中で、最初に「0x000」でない飛び先をその真の飛び先と判断する。即ち、本実施形態では、飛び先リスト73を重ね書きすることで修正プログラムの存在を明らかにし、この修正プログラムに制御を移すことでアプリケーションプログラムの修正を実現している。図10 (B) の

飛び先リスト73においては、飛び先は3個提示できるから、修正は各個別プログラム74につき2回までできることになる。

#### 【0057】

次に、図10（B）の飛び先リストL2を見てみる。個別プログラムK2の場合、その飛び先は飛び先リストL2で判断され、この第1のアドレスは「0x000」であり、この第2のアドレスは（A2）であり、この第3のアドレスは「0x7fff」である。第1のアドレスが「0x000」であることは、既に重ね書きされて他の修正プログラムが存在することを意味する。その第2のアドレスが（A2）であり、これが「0x000」でないことは、これが個別プログラムK2の修正プログラムの飛び先であることを意味している。第3のアドレス「0x7fff」の意味するところは、これらのビットにおける浮遊ゲートが紫外線照射等で帶電した後1度も書き込みされていないことである。これは次の修正プログラムのためのアドレスであり、いわば予備アドレスである。最後に飛び先リストL<sub>n</sub>を見てみる。第1と第2のアドレスは「0x000」であり、第3のアドレスは（An）であるから、既に2度の修正をして個別プログラムK<sub>n</sub>の修正プログラムはアドレス（An）に存在することになる。

#### 【0058】

この第4の実施形態では、次のような効果がある。

本実施形態では、アプリケーションプログラムの構造が予めかなり制限されたものである。つまり、メインプログラム71の他に、飛び先判定プログラム72、構造化された飛び先リスト（L<sub>1</sub>, …, L<sub>n</sub>）73、及び機能化された個別プログラム（K<sub>1</sub>, …, K<sub>n</sub>）74を有し、さらにこの他にEPROM12のアドレス空間内に修正プログラム領域（S<sub>1</sub>, …, S<sub>m</sub>）75を持つ余裕がなければならない。しかし、本実施形態では、図1及び図3の検出手段18を必要とせず、専らアプリケーションプログラムの構造によってプログラムの修正を可能とするものであって、何ら付加すべき電子回路のハードウェアを必要としない点で優れている。

#### 【0059】

なお、本発明は上記実施形態に限定されず、例えば、図3のOTPマイコンを

他の構成に変更したり、EEPROM12を他の再書き込み可能な不揮発性メモリで構成したり、あるいはEEPROM12に格納されるプログラムを図示以外の構成に変更する等、種々の変形が可能である。

## 【0060】

## 【発明の効果】

以上詳細に説明したように、第1及び第2の発明によれば、CPUによる実行プログラムの実行時に、プログラム領域中のプログラム消滅領域を検出手段で検出して割り込み信号を出力して該CPUに与え、割り込みプログラムに従い、該実行プログラムの実行を中断して修正プログラムを実行し、修正プログラムの終了後に実行プログラムへ復帰させるようにしたので、何ら外付けの修正手段を使わず、かつ、OTPマイコンの利便性を損なうことなく、アプリケーションプログラムの複数回の修正を可能にし、これに要するコストも若干の回路付加で済ませることができる。

## 【0061】

第3の発明によれば、予め定められた固定アドレス格納領域に、検出すべきプログラム消滅領域のアドレスを書き込むことで、検出手段によって修正プログラムの存在を検知し、かつ、CPUに対し割り込みを生じさせるようにしたので、実行プログラムの初期化段階でプログラムの修正を前提とした初期化を行っておけば、その後は複数回の修正プログラムの追加に対応することができる。

## 【0062】

第4の発明によれば、不揮発性メモリに固定アドレス格納領域及び飛び先判定プログラムを設け、CPUによる実行プログラムの実行時に、実行命令コードと予め設定された指定命令コードとを比較手段で比較し、両者が一致したときには割り込み信号を出力し、この割り込み信号によってCPUが割り込みプログラムに従い、実行プログラムの実行を中断して飛び先判定プログラムへ移り、固定アドレス格納領域を参照して修正プログラムの位置を判定し、修正プログラムを実行してこの終了後に実行プログラムへ復帰するようにしたので、簡単な電子回路等のハードウェアを付加するだけで、複数回のプログラムの修正が可能になる。

## 【0063】

第5の発明によれば、C P Uがメインプログラムを実行するときには、飛び先判定プログラムに応じて飛び先リストを参照し、修正プログラムがない場合には該当する個別プログラムを実行し、修正プログラムがある場合には該当する修正プログラムを実行するようにしたので、何ら電子回路等のハードウェアを付加することなく、専ら不揮発性メモリに格納されるプログラムの構造によってこのプログラムの複数回の修正が可能になる。

第6の発明によれば、不揮発性メモリをE P R O Mで構成したので、このE P R O Mの重ね書きができる点に着目して簡単かつ的確に複数回のプログラムの修正が可能になる。

【図面の簡単な説明】

【図1】

本発明の実施形態の原理を示す概念図である。

【図2】

従来のO T Pマイコンを示す一般的な構成図である。

【図3】

本発明の実施形態を示すO T Pマイコンの概略の構成図である。

【図4】

図3中のC P Uの概略を示す構成図である。

【図5】

図3中の検出手段を示す構成図である。

【図6】

本発明の第2の実施形態を示す検出手段の他の構成図である。

【図7】

本発明の第2の実施形態を示すE P R O Mのアドレス空間の図である。

【図8】

本発明の第3の実施形態を示す検出手段の他の構成図である。

【図9】

本発明の第3の実施形態を示すE P R O Mのアドレス空間の図である。

【図10】

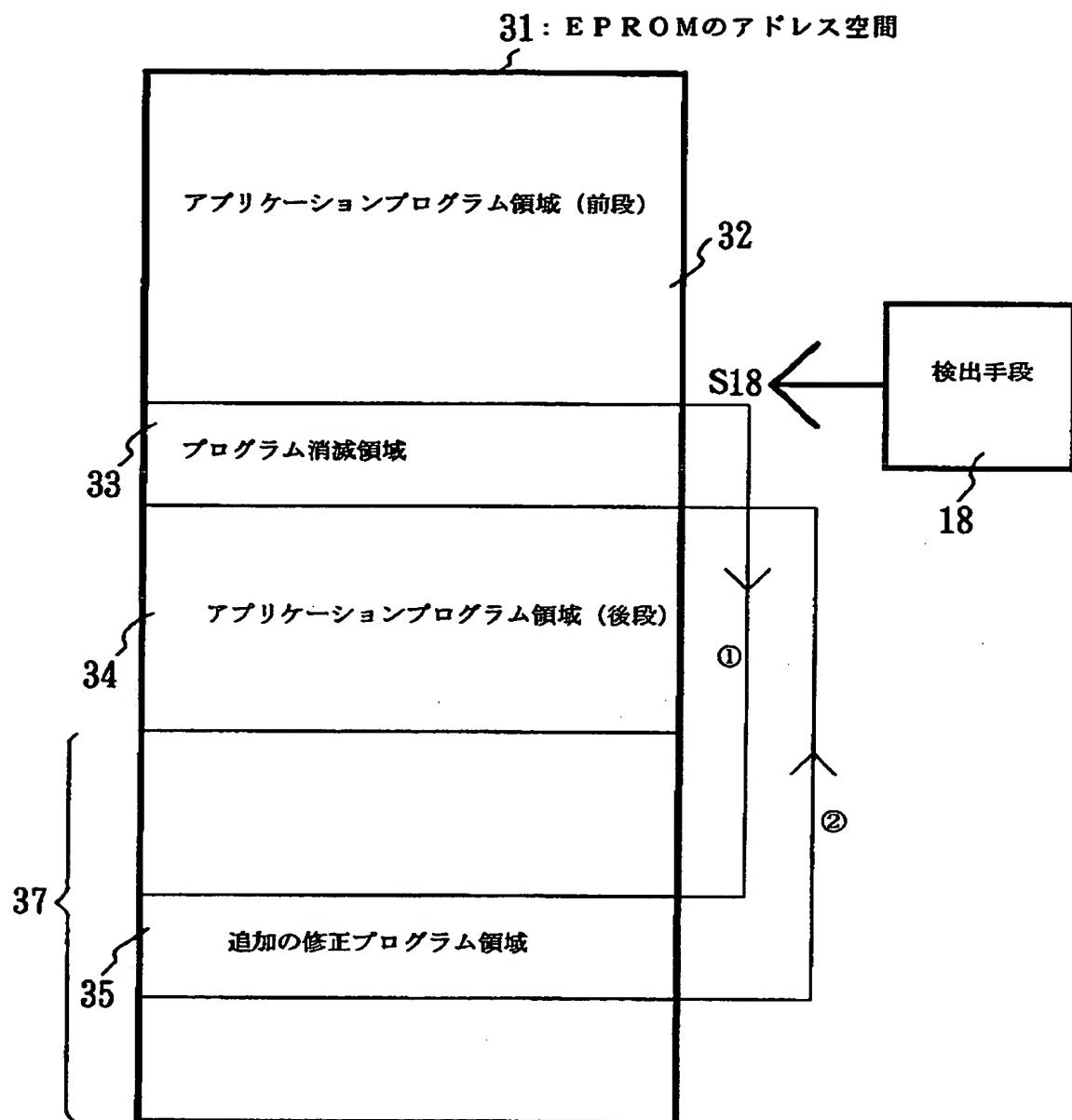
本発明の第4の実施形態を示すアプリケーションプログラムのプログラム構造図である。

【符号の説明】

- 1 2 E P R O M
- 1 3 E P R O M書き込み回路
- 1 4 高電圧端子
- 1 5 R A M
- 1 7 C P U
- 1 8 検出手段
- 3 1 E P R O Mのアドレス空間
- 3 2, 3 4 アプリケーションプログラム領域
- 3 3 プログラム消滅領域
- 3 5 追加の修正プログラム領域
- 4 3 比較回路
- 4 5 カウンタ
- 4 6, 4 9 アドレスレジスタ
- 5 1, 6 2 固定アドレス格納領域
- 6 1 飛び先判定プログラム
- 7 1 メインプログラム
- 7 2 飛び先判定プログラム
- 7 3 飛び先リスト
- 7 4 個別プログラム
- 7 5 修正プログラム領域

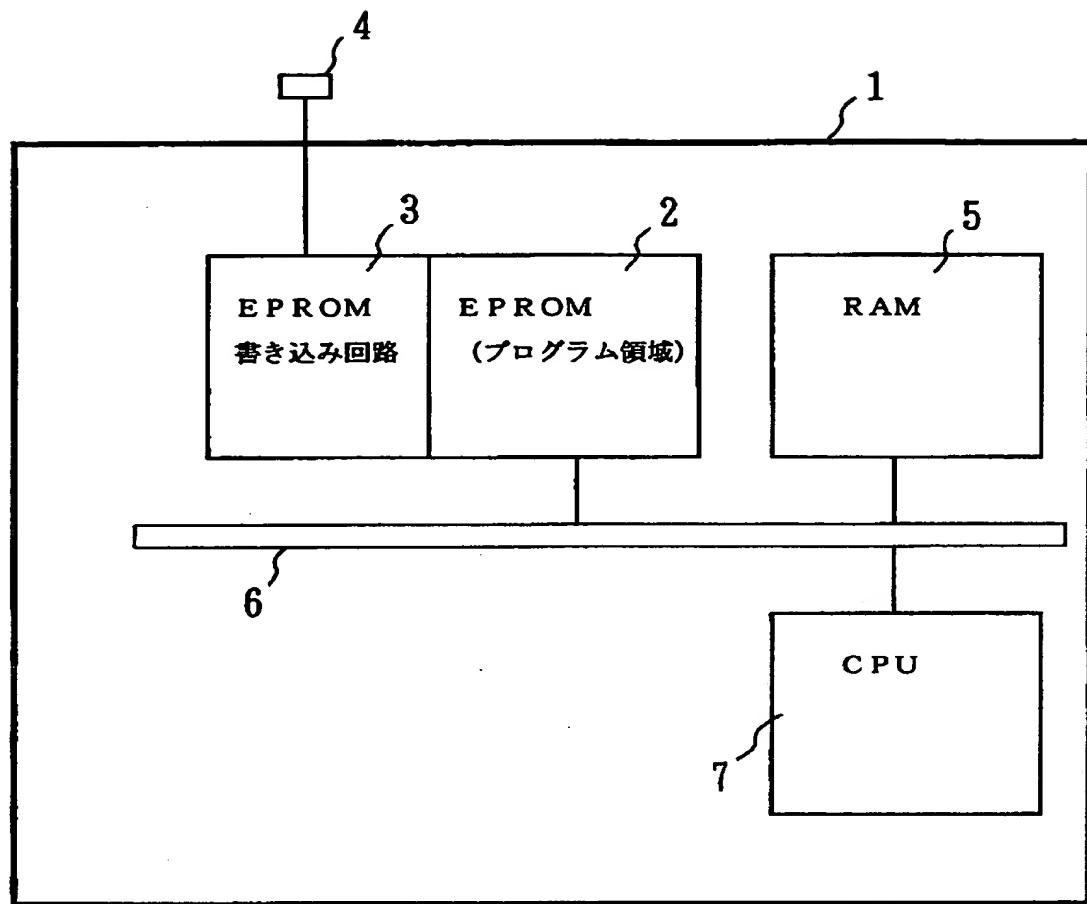
【書類名】 図面

【図1】

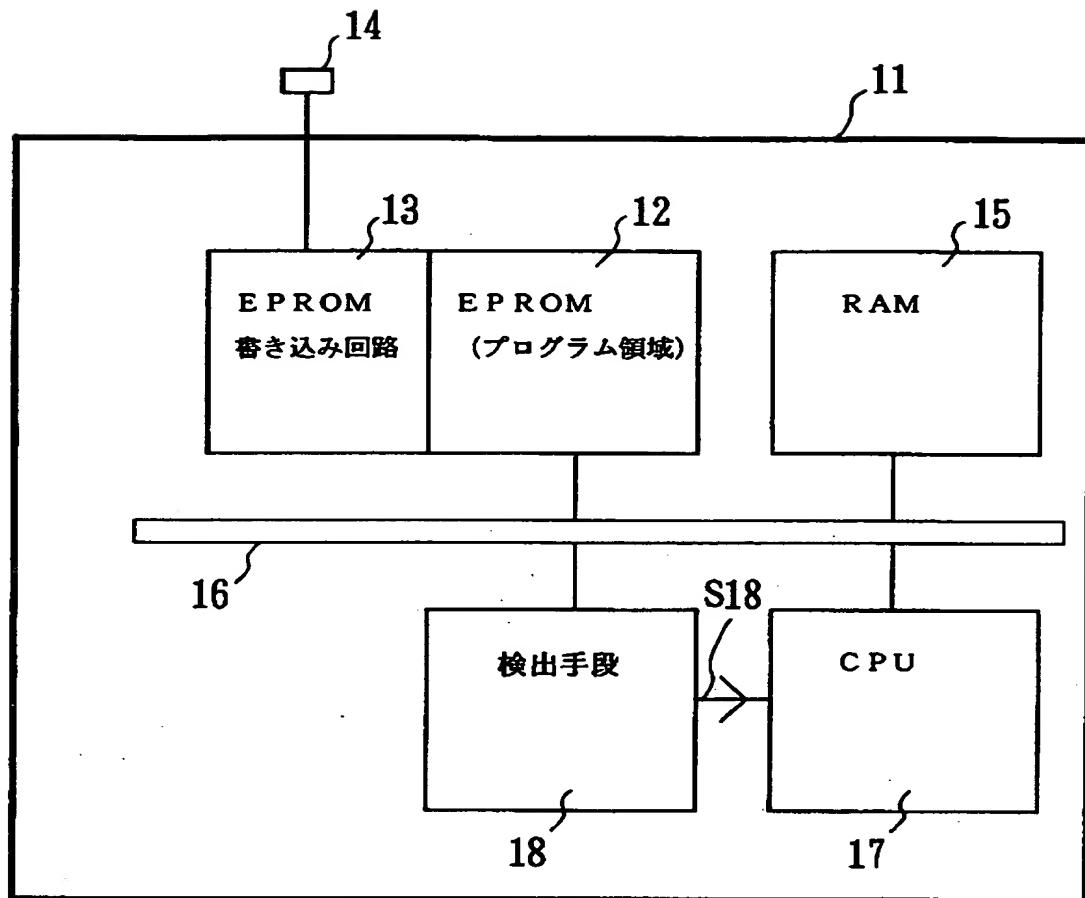


本発明の実施形態の原理

【図2】



【図3】



本発明の実施形態のOTPマイコン

【図4】

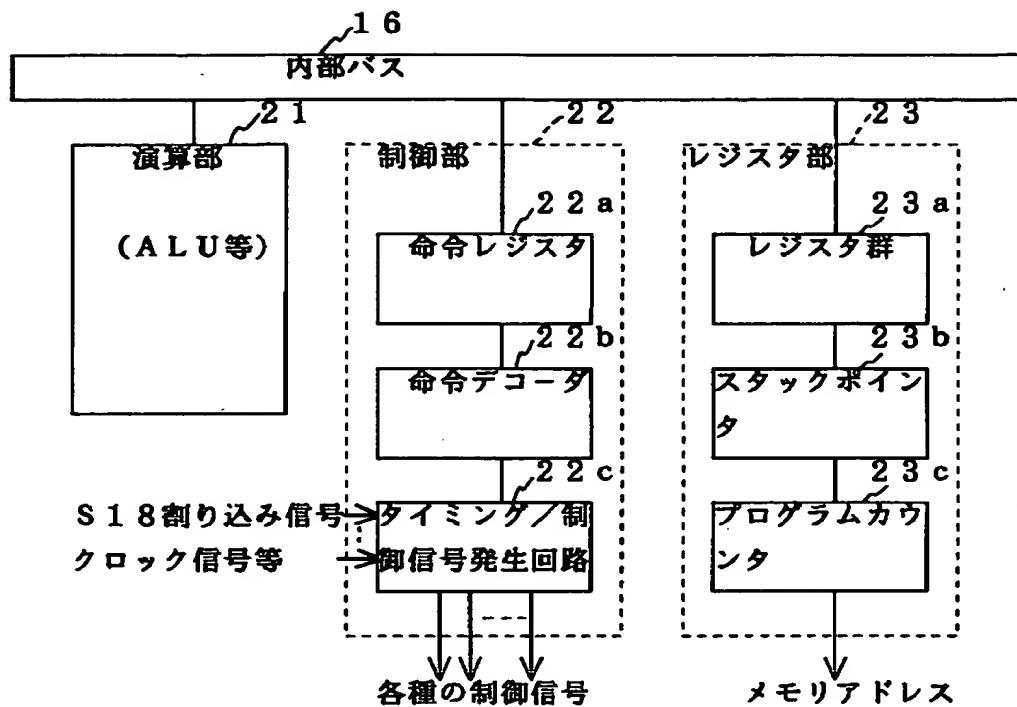


図3中のCPU

【図5】

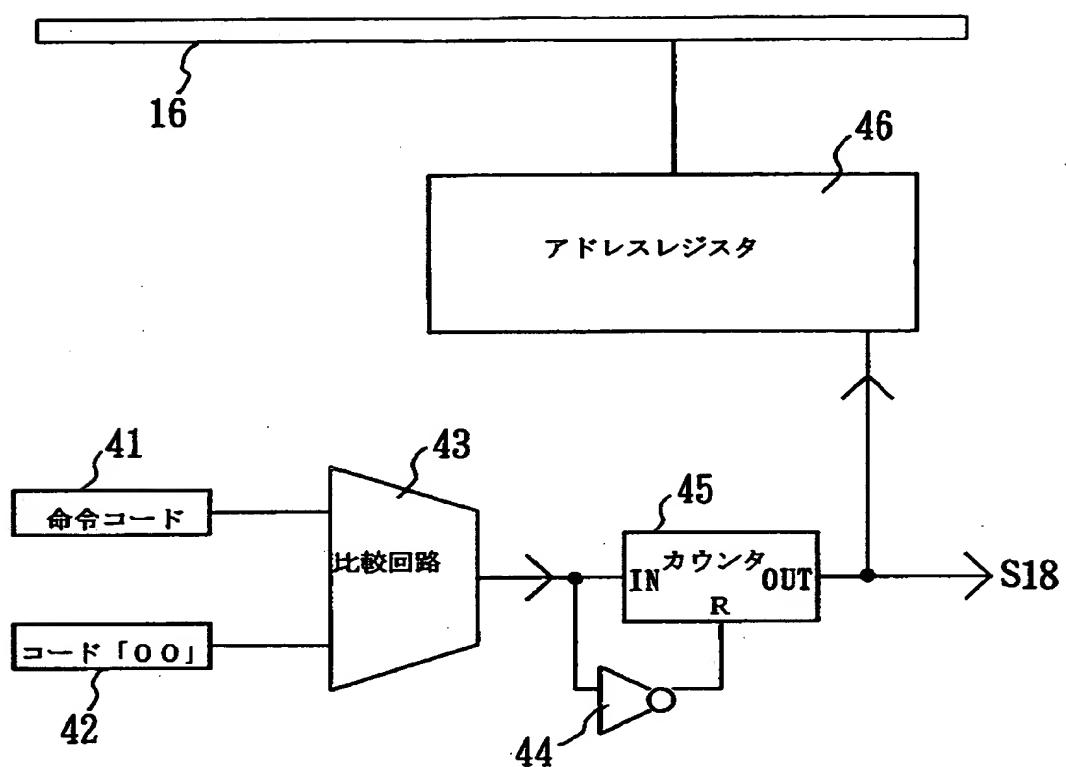
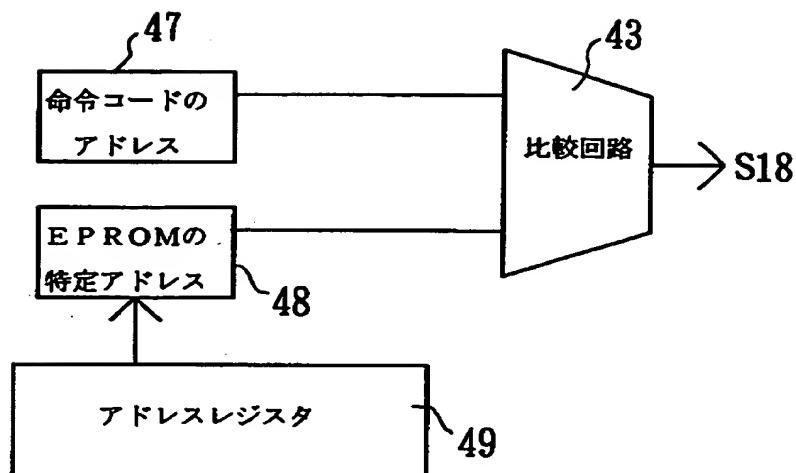


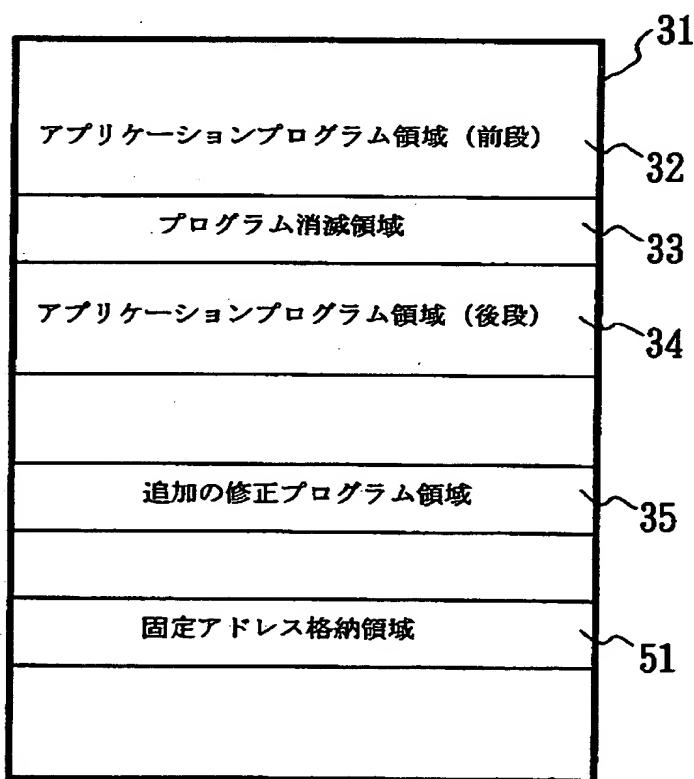
図3中の検出手段の構成

【図6】



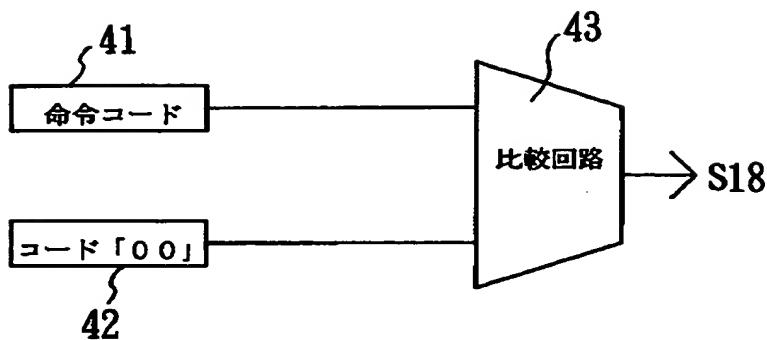
本発明の第2の実施形態の検出手段の他の構成

【図7】



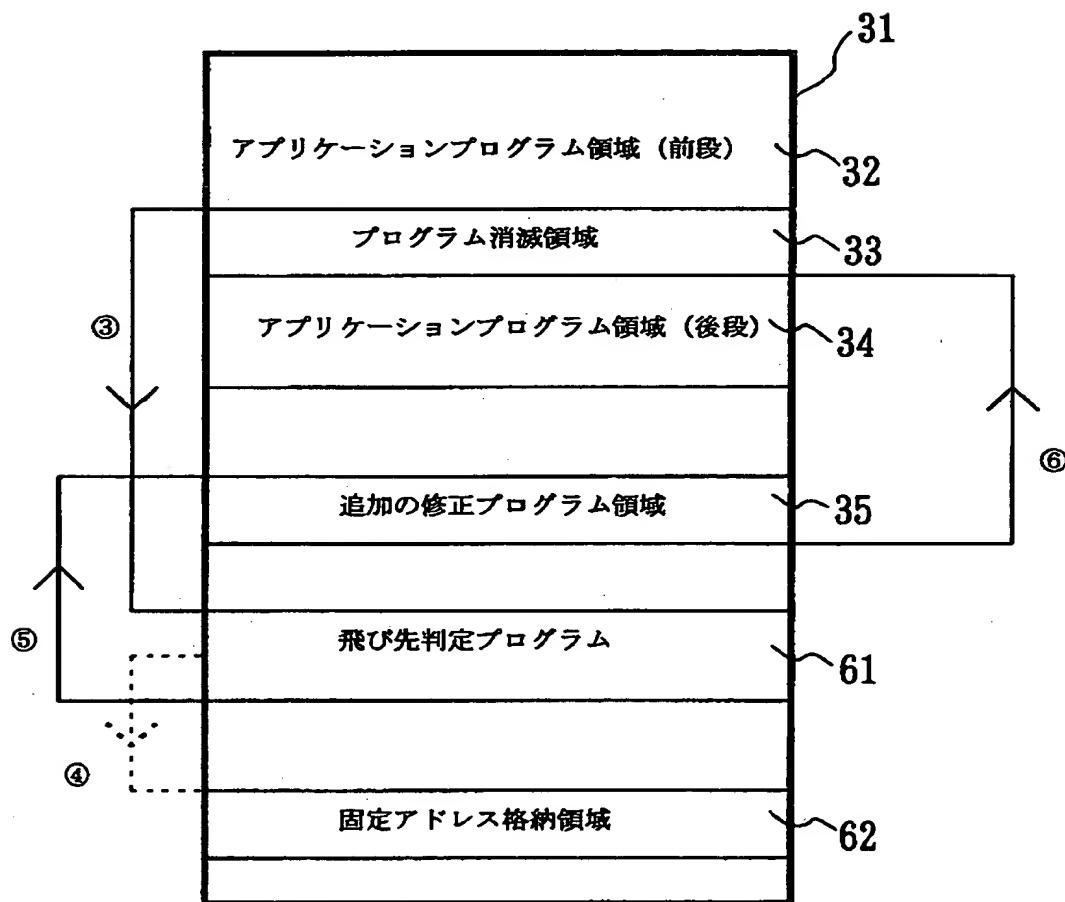
本発明の第2の実施形態のアドレス空間

【図8】



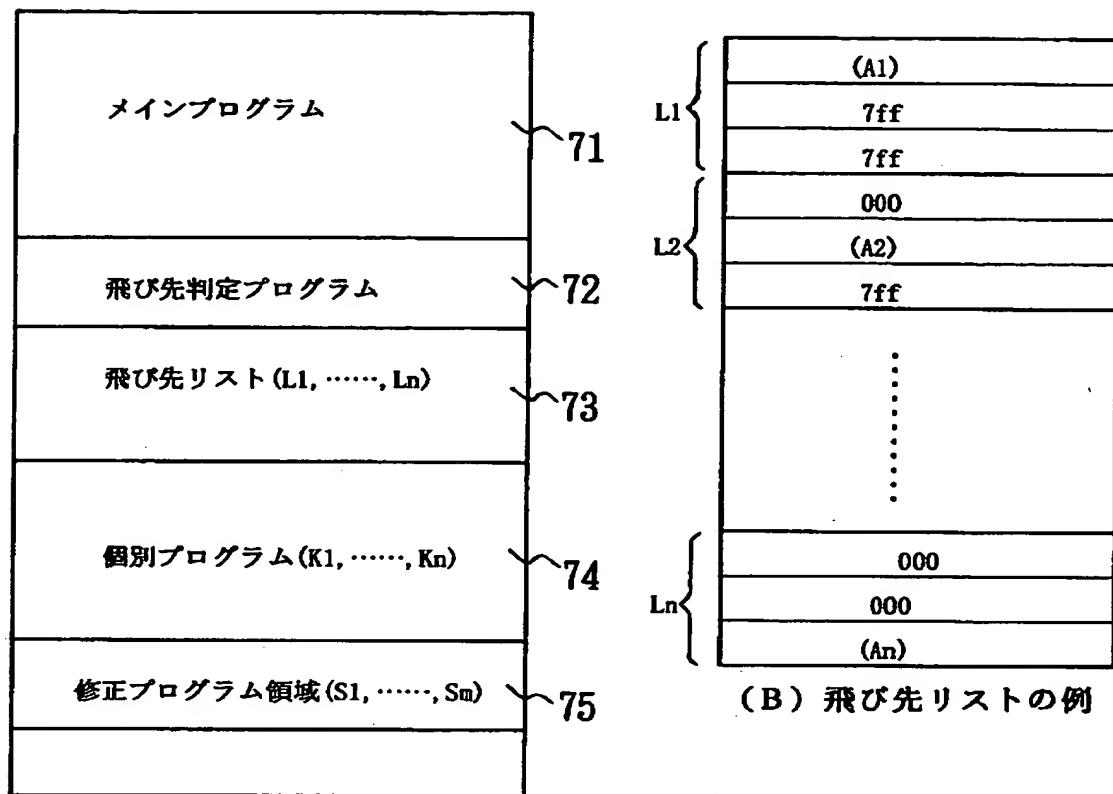
本発明の第3の実施形態の検出手段の他の構成

【図9】



本発明の第3の実施形態のアドレス空間

【図10】



本発明の第4の実施形態のプログラム構造

【書類名】 要約書

【要約】

【課題】 複数回プログラム変更できるOTPマイクロコントローラ (One Time Programmable MicroController) を提供する。

【解決手段】 E PROMのアドレス空間31において、アプリケーションプログラム領域32、34の一部を修正する場合には、この修正すべき部分の先頭の一部又は全部を書き込み状態のゲートに変えたプログラム消滅領域33を作り、修正すべき追加の修正プログラム領域35を設ける。CPUによって領域32のアプリケーションプログラムを実行し、検出手段18によって領域33を検出すると、この検出手段18から割り込み信号S18が出力されてCPUに与えられる。CPUは、割り込みプログラムに従い、領域35へジャンプし、このプログラムを実行する。修正プログラムが終了すると、CPUは、割り込みプログラムに従い、領域34へ戻り、このプログラムを実行する。

【選択図】 図1

出願人履歴情報

識別番号 [000000295]

1. 変更年月日 1990年 8月22日

[変更理由] 新規登録

住 所 東京都港区虎ノ門1丁目7番12号  
氏 名 沖電気工業株式会社